Kontakt z mikroinformatyką można zacząć różnie. Niektórzy znacznym nakładem finansowym sprowadzają z zagranicy gotowy mikrokomputer. Inni składają go sami, ponosząc wprawdzie mniejsze koszty, ale borykając się z większymi trudnościami. MINI-MONITOR jest propozycją właśnie dla tych drugich. Zawiera on stosunkowo niewiele funkcji, ale zajmuje niewiele ponad 800 bajtów. Zapisany w pamięci EPROM (wystarczy tu jeden układ 2708), może oddać nieocenione usługi przy oprogramowywaniu własnej konstrukcji.

MINI-MONITOR wymaga dołączenia dwóch procedur — wprowadzania i wyprowadzania znaku na konsolę operatorską. Są one ściśle zależne od zastosowanej konstrukcji i przy braku standardowych rozwiązań nie sposób przewidzieć wszystkich możliwych opcji. Warto więc podkreślić, że autor przygotował MINI-MONITOR tak, aby można go było wykorzystać w dowolnej konstrukcji.

MINI-MONITOR dla 8080 i Z80

Przedstawiony poniżej program realizuje pięć funkcji. Umożliwia przeglądanie i zmianę zawartości pamięci oraz pozwala na uruchamianie programów w języku maszynowym mikroprocesorów 8080 i Z80.

MINI-MONITOR umieszczony jest w pamięci od adresu 0. Umożliwia to rozpoczęcie realizacji programu przez podanie sygnału zerującego RESET. Pojawienie się na konsoli operatora znaku ">" oznacza gotowość programu do pracy (wykonywania poleceń). Polecenie składa się z kodu polecenia (litera: M, H, A, G lub B), jednego lub dwóch argumentów i znaku CR (powrót karetki, w kodzie ASCII--0DH). Argumenty podaje się w kodzie szesnastkowym. W przypadku podawania dwóch argumentów, należy je rozdzielić co najmniej jedną spacją (odstępem).

Polecenie:

M — przeglądanie i zmiana zawartości pamięci.
M adres [CR]

Na konsolę wyprowadzony zostaje adres i zawartość komórki pamięci o podanym adresie. Program oczekuje teraz na wprowadzenie znaku (znaków). Po wprowadzeniu kodu CR, program powtarza działanie dla kolejnej komórki pamięci (p. przykład 1). Aby zmienić zawartość komórki, należy wprowadzić dwie cyfry szesnastkowe, a następnie CR (p. przykład 2).

Wprowadzenie kropki "." kończy wykonywanie polecenia.

H — wyprowadzenie zawartości obszaru pamięci (szesnastkowo) na konsolę operatora.
H adres1 adres2 [CR]

Zawartość obszaru pamięci zostaje wyprowadzona z przestrzeni pomiędzy adres1 ... adres 2 w wierszach po osiem bajtów w każdym (p. przykład 3).

A — wyprowadzenie zawartości obszaru pamięci (znaki ASCII)
A adres1 adres2 [CR]
Jak wyżej. Wartości spoza tablicy kodów ASCII reprezentuje kropka "." (p. przykład 4).

G — rozpoczęcie realizacji programu
G adres [CR]
Rozpoczyna wykonywanie programu użytkownika od podanego adresu.

B — ustawienie lub usunięcie pułapki
B adres [CR]
Do pamięci, pod wskazanym adresem, wstawiona zostaje instrukcja CALL BR. Napotkanie tej instrukcji w czasie wykonywania programu powoduje wyprowadzenie zawartości rejestrów w postaci: R W1 (W2). R oznacza tu nazwę pary rejestrów (AS to odpowiednik PSW w 8080, ST to dwa bajty z wierzchołka stosu, czyli adres instrukcji po CALL BR), W1 — zawartość rejestrów, a W2 — zawartość komórki pamięci wskazanej przez adres znajdujący się w parze rejestrów (p. przykłady 5 i 6).
B [CR]

Wstawia na miejsce CALL BR poprzednią zawartość (p. przykład 7). Nie

Przykład 1. Przeglądanie zawartości pamięci

```
>M1000[CR]
1000 06[CR]
1001 41[CR]
1002 3E[CR]
1003 42[CR]
1004 21[CR]
1005 01[CR]
1006 44 .[CR]
>
```

Przykład 2. Zmiana zawartości komórek pamięci

```
>M1000[CR]
1000 20 06[CR]
1001 46 41[CR]
1002 49 3E[CR]
1003 4C 42[CR]
1004 45 21[CR]
1005 20 01[CR]
1006 52 44[CR]
1007 45 C3[CR]
1008 41 00[CR]
1009 44 20[CR]
100A 20 .[CR]
>
```

Komputerku, jesteś wolny!

Prezes Głównego Urzędu Ceł do profesora Iwo Białynickiego-Biruli (Zakład Fizyki Teoretycznej PAN):

W związku z pismem z 7 listopada 1981, proponującym zmianę zasad przywozu z zagranicy komputerów „osobistych", oraz nawiązując do poprzedniej korespondencji w tej sprawie, uprzejmie informuję Obywatela Profesora, że poleciłem urzędom celnym wydawanie „z urzędu", tzn. w trakcie odprawy celnej i bez żądania wniosków zainteresowanych, pozwoleń na przywóz z zagranicy komputerów „osobistych" i „domowych".

Przyjęte rozwiązanie jest jedynym możliwym w obowiązującym stanie prawnym, gdyż objęcie ewentualnie komputerów poz. 58 pkt 4 taryfy celnej przywozowej nie jest możliwe ze względów formalnych (Systematyczny Wykaz Wyrobów, na którym oparta jest nomenklatura taryfy celnej, zalicza wszelki sprzęt komputerowy do branży 092).

Polecone zasady będą obowiązywały przejściowo do czasu zmiany rozporządzenia Ministra Handlu Zagranicznego ustanawiającego taryfę celną.

Jakież to niewiarygodne! Jeszcze niedawno powoływano się w Urzędzie Ceł na dobro państwa — zmuszonego bronić się przed komputerami o niezbadanych możliwościach... A tu naraz: proszę, droga wolna. I — co więcej — jest to droga najwygodniejsza; cło pozostaje bowiem na dotychczasowym, niskim poziomie. Doprawdy, zadziwiające zwycięstwo rozsądku.

Tak czy inaczej, wydarzenie jest szczęśliwe i nie ma co kręcić głową. Rozważmy raczej możliwe następstwa. Skoro procedura celna została uproszczona, spodziewać się teraz można znacznego dopływu sprzętu. Nie tylko przywożonego przez Polaków z zagranicy, ale też przysyłanego stamtąd. Być może nadawcami będą także organizacje (np. polonijne), dla których rozwój cywilizacyjny Polski nie jest sprawą bez znaczenia. Pomyślmy więc — jeśli zdarzy się tak, w jaki sposób zdołamy to przyjąć. Kto i na jakiej zasadzie? Czy nie jest to dobry moment na dokonanie dalej idącej samoorganizacji amatorów mikroinformatyki.

Zresztą, na zorganizowanie się samemu każdy moment jest najlepszy.

ZBIGNIEW GLUZA

Przykład 3. Wyprowadzenie zawartości pamięci (szesnastkowo)

```
>H1000 100B[CR]
1000 06 41 3E 42 21 01 44 C3
1008 00 20 20 43 44 45 46 47
>
```

Przykład 4. Wyprowadzenie zawartości pamięci (znaki ASCII)

```
>A1000 100B[CR]
1000 . A>B ! . D .
1008 . C D E F G
>
```

Przykład 5. Ustawienie pułapki (punktu wstrzymania realizacji programu)

```
>B1004[CR]
>M1004[CR]
1004 CD[CR]
1005 16[CR]
1006 01 .[CR]
```

Przykład 6. Rozpoczęcie realizacji programu i wyprowadzenie zawartości rejestrów po wstrzymaniu spowodowanym napotkaniem pułapki

```
>G1000[CR]
AS 4200 (C1)
BC 410F (45)
DE 1000 (06)
HL 03AE (10)
ST 1007 (C3)
```

Przykład 7. Usunięcie pułapki

```
>B[CR]
>M1004[CR]
1004 21[CR]
1005 01[CR]
1006 44[CR]
1007 C3 .[CR]
```

można ustawić jednocześnie więcej niż jednej pułapki.

•

MINI-MONITOR może pracować na dowolnym mikrokomputerze wykorzystującym mikroprocesor 8080 lub Z80.

Kod programu zajmuje stałe miejsce w pamięci (ROM lub RAM) od adresu 0 do 29FH. (Kolejne bajty podane na wydruku należy umieścić w pamięci poczynając od adresu 0). Program wymaga 48 bajtów pamięci RAM; 24 bajty — obszar roboczy, 24 — stos. Początkowy adres obszaru roboczego należy podać w komórkach: 29DH — bajt mniej znaczący adresu i 29EH — bajt bardziej znaczący. Adres ten będzie jednocześnie wykorzystany przy ustaleniu wskaźnika stosu.

Kod MINI-MONITORA wymaga uzupełnienia o procedurę wprowadzania znaku z konsoli (klawiatury) do akumulatora — CZYT oraz procedurę wyprowadzania znaku z akumulatora na konsolę (ekran monitora) — **PISZ.** Procedury takie zależą od konstrukcji mikrokomputera. Adresy procedur należy umieścić w programie pod adresami podanymi w tabeli:

Na wydruku przedstawiono kod programu MINI-MONITOR w postaci

Procedura	Bajt mniej znaczący	Bajt bardziej znaczący
PISZ	28CH	28DH
CZYT	297H	294H

```
A)type mintmon.hex
:100000002A9D02F90610AF772305C20600CD08022F
:10001000E3ECD8702CD45022A9D02EB214500AF31
:10002000AFE0DCA0D008ECA3600EFFBECA0D0044
:10003000232323C32000E5CDC001E1235E23562103
:10004000D00E5D5C94D8F00475500485E0042C8F8
:10005000004156B1FFAFCD6B025E2356D5C9AFCD2F
:10006000B02CD7F02CD0802CD17020608CD120229
:10007000ECDF1012303C26D00E5AFCD6902CD7FD4
:10008000020EBE1E5CD7502E1D8CD0802C36800AF0F
:10009000CD6B02CD7F02CD0802CD1702CD12027EBC
:1000A000CDF101CD1202E5CD4502AF2A9D02EBE173
:1000B0001AFE0DC2BA0023C39600FE2ECBE5CD9CE1
:1000C000017CE17723C39600AF2A9D02237EFE0DBB
:1000D000CAFF00AFCD6502E5AFCD6B025E2356EBE4
:1000E000D1E506037E12231305C2E400AFCD6702FB
:1000F000D1732372EB36CD231117017323729C9AF6D
:10010000CD67025E2356D5AFCD6502D106037E12C0
:10011000231305C20E01C9E5D5C5F5CD0802E106D8
:1001200041 0E53CD2002CD2B02E106420E43CD20DD
:10013000020D2B02E106440E45CD2002CD2B02E17B
:10014000064B0E4CCD2002CD2B02E106530E54CDB5
:10015000202CD2B02C9AFCD6B02CD7F02CD0B02AC
:10016000CD17020608CD1202AF7EDE203E2EDA7DCC
:10017000017EDE7ADA7C013E2EC37D017ECDB702D0
:10018000023AF05C26501E5AFCD6902CD7F02EBE1BA
:10019000E5CD7502E1DBCD0B02C36001F5C52100A7
:1001A0000006041AD630FE0AFAAD01D607E60FB5EE
:1001B0006F1305CABD0129292929C3A301C1F1C9AA
:1001C0002A9D0223EBCDE801CD9C01D5E5AFCD6B97
:1001D00002D1732372D1CDE801CD9C01D5E5AFCD1D
:1001E0006902D1732372D1C9AF1AFE20C013C3E8CC
:1001F00001F51F1F1F1FCDFA01F1E60FC630FE3AB1
:1020000DA0502C607C3B7023E0DCDB7023E0AC348
:10021000B7023E20C3B7027CCDF1017DCDF101C96B
:1002200078CDB70279CDB702C31202CD1702CD1295
:10023000023E28CDB7027ECDF1013E29CDB702CD39
:10024000120 2C308022A9D02AFCD9302CDB7027726
:1002500023FE0BC25F022B36202B3620C34802FE45
:10026000DC8C34802C602C602C602C60F4F06002A
:100270002A9D0209C9237CB537C8B7B957A9CC97E23
:10028000F5237E67F16FC9F5E5D5C5CD0000C1D175
:0F029000E1F1C9E5D5C5CD0000C1D1E1C900003C
:0000000000
```

Wydruk kodu programu MINI-MONITOR w formacie INTEL-HEX (ostatni bajt w wierszu jest bajtem kontrolnym).

szesnastkowej. Wersję źródłową w języku ASSEMBLER 8080 można otrzymać w firmie CSK.

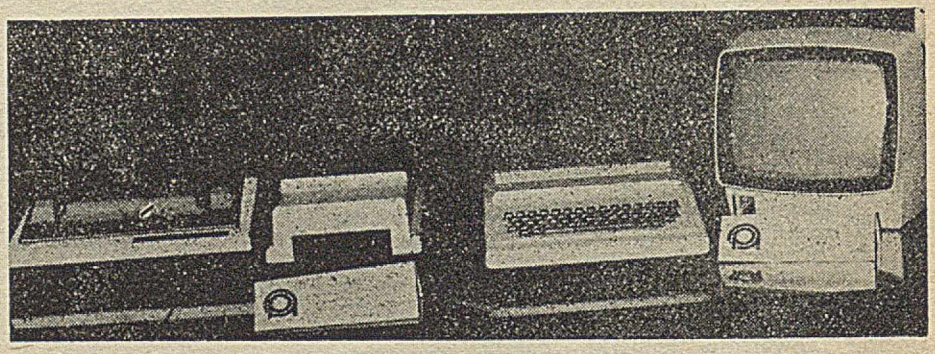
JERZY DWORZECKI
Computer Studio Kajkowski
Gdynia

Zgodnie z obietnicą, zamieszczamy opis kolejnego mikrokomputera produkowanego w kraju. W porównaniu do ZX81 jest to niewątpliwie rozwiązanie o klasę lepsze. Niestety, również sporo droższe.

AMEPROD ocenia możliwości produkcyjne na około trzysta egzemplarzy rocznie. Nie trudno więc wyrokować, że okaże się to kroplą w morzu potrzeb. Tym bardziej, że krążą wieści o podejmowanej przez PKP próbie zakupienia... tysiąca egzemplarzy. W krajach zachodnich duże zamówienie to olbrzymia szansa dla producenta. U nas może oznaczać... kilkuletnie zniknięcie firmy z rynku. Przyszłość pokaże, czy AMEPRODOWI uda się pogodzić ambicje upowszechniania mikroinformatyki z „propozycjami nie do odrzucenia".

Komputer osobisty AC 805

Doświadczenie zebrane przez ponad dwa lata produkcji i promocji komputera osobistego ZX81 skłoniły Przedsiębiorstwo Zagraniczne AMEPROD do podjęcia prac nad własnymi systema-



mi komputerowymi. Prezentowany komputer **AC 805**, którego pierwszy pokaz odbył się na MTP w 1984 roku, jest wersją podstawową w serii **AC 800**, w skład której wchodzą ponadto modele: **AC 815** oraz **AC 825.**

W zwartej obudowie, o ładnym, opływowym kształcie, umieszczono płytę CPU, płytę klawiatury, a także zasilacz dostarczający niezbędnych napięć dla systemu. Rolę jednostki centralnej spełnia 8-bitowy mikroprocesor Z80. Minimalny system obliczeniowy składa się z AC 805, monitora ekranowego lub telewizora (pracującego jako monitor ekranowy) i magnetofonu kasetowego (pamięć zewnętrzna). AC 805 został wyposażony w **Mikro System,** interpreter języka **BASIC** (zgodny ze standardem MICROSOFT, zbliżony do wersji zastosowanej w NASCOM COMP.) oraz **edytor ekranowy.** Wymienione oprogramowanie systemowe rezyduje w pamięci stałej ROM, o pojemności 12 KB. Przewidziano trzy wersje pamięci dynamicznej RAM o pojemnościach: 16, 32 oraz 48 KB.

Uruchomienie AC 805 jest bardzo proste — sprowadza się do połączenia go z monitorem i magnetofonem oraz włączenia do sieci. Po włączeniu zgłasza się Mikro System. Użytkownik ma do dyspozycji 19 dyrektyw, które umożliwiają:

— wejście do interpretera BASICA — „zimne"

MIKROKOMPUTER AC 805

jednostka centralna: Z80
oprogramowanie:
- interpreter języka BASIC według standardu MICROSOFT (8 KB)
- MONITOR o nazwie Mikro System
- opcjonalnie system operacyjny kompatybilny z CP/M 2.2

pamięć ROM — 12 KB
pamięć RAM — 16, 32 lub 48 KB
pamięć masowa:
- standardowo magnetofon (modulacja FSK 300 lub 1200 bodów)
- opcjonalnie dyski elastyczne 5,25″ — 1 MB
- opcjonalnie twarde dyski (WINCHESTER) — 10 lub 30 MB

wyświetlanie: 24 linie po 32 znaki, 96 znaków ASCII, 32 znaki graficzne, 128 znaków programowalnych, możliwa pseudografika
sprzęgi: szeregowy — 2 × RS 232C, 150... 4800 bodów; równoległy — 24 linie TTL
klawiatura: QWERTY; mata gumowa, plastikowe klawisze
dokładność obliczeń: 7 cyfr
cena: model podstawowy — 400 tys. zł (sierpień 1984)
producent: AMEPROD, 61-632 Poznań, Kmieca 20A, tel. 22 18 79
termin realizacji zamówienia: zależy od liczby zamówień, produkcja w 1985 roku — ok. 300 egzemplarzy

— wejście do interpretera BASICA — „gorące"
— zmianę konfiguracji wejść i wyjść
— rozpoczęcie realizacji programu w kodzie maszynowym
— zapis i odczyt programów przy współpracy z magnetofonem
— zmianę szybkości transmisji portów szeregowych
— ustawienie punktu wstrzymania (pułapki) dla wykonywanego programu
— kontrolę i aktualizację zawartości poszczególnych lokacji pamięci i rejestrów.

AC 805 został zaprojektowany jako komputer osobisty przeznaczony do zastosowań profesjonalnych, który z powodzeniem może pełnić rolę pomocy dydaktycznej lub narzędzia wspomagającego inżyniera. Z tego powodu szczególnie starannie opracowano komunikację komputera z urządzeniami zewnętrznymi. AC 805 ma dwa porty szeregowe (w standardzie RS 232 C) AUX i HOST, o ustawianej prędkości transmisji w zakresie 150—4800 bodów, oraz 24-bitowy programowalny port równoległy. Transmisja szeregowa może odbywać się bez udziału jednostki centralnej, dzięki czemu istnieje możliwość łączenia komputerów AC 850 ze sobą, przy jednoczesnej transmisji pomiędzy komputerem a dołączonym do portu szeregowego urządzeniem zewnętrznym. Port równoległy umożliwia dołączenie do systemu nietypowych urządzeń peryferyjnych, np. rejestratorów, sterowników czy drukarek i ploterów współpracujących poprzez złącze równoległe.

Klawiatura w układzie typu QWERTY została zbudowana z maty gumowej i nakładek z tworzywa sztucznego. Zastosowanie maty z histerezą mechaniczną zapewnia użytkownikowi jednoznaczną informację o naciśnięciu klawisza. Klawisze specjalne (np. CLEAR, HOME) znacznie ułatwiają pisanie programów i ich redagowanie. Współpraca z monitorem odbywa się poprzez wyjście video lub modulator UHF, pracujący na 36 kanale. Układ obsługi monitora ekranowego wyświetla 24 wiersze po 32 znaki i działa na zasadzie pamięci obrazowej (ang. video-RAM). Pamięć obrazowa jest adresowana jak pamięć mikrokomputera.

Mikrokomputer wyposażony jest w dwa generatory znaków. Generator podstawowy zawiera pełen zestaw znaków kodu ASCII, z dużymi i małymi literami oraz kilkunastoma znakami graficznymi. Rezyduje on w pamięci ROM. Drugi — dodatkowy generator — pozwala na zdefiniowanie 128 znaków przez użytkownika. Dzięki temu możliwe jest uzyskanie pseudografiki w całym obszarze obrazu, a także tworzenie nietypowych znaków alfabetu lub też drugiego, alternatywnego alfabetu (np. cyrylicy). Dostęp do pamięci obrazowej nie jest zsynchronizowany z sygnałami wygaszania plamki [1]).

Podkreślając walory dydaktyczne systemu, należy zwrócić uwagę na możliwość uruchamiania programów (pod kontrolą Mikro Systemu) wprowadzonych w kodzie maszynowych ich redagowania oraz współpracy z programami napisanymi w języku BASIC.

Pisanie i uruchamianie programów w BASICU wydatnie ułatwia edytor ekranowy z kursorem poruszającym się po ekranie we wszystkich kierunkach — z możliwością usuwania i wstawiania linii oraz wymiany zawartości linii programu.

Unikalną cechą AC 805 jest możliwość przygotowania programów w kodzie maszynowym dla zestawu laboratoryjnego ZLA 01 [2]), a wynika to z przyjęcia takiego samego standardu zapisu na taśmie magnetofonowej w obu urządzeniach (modulacja FSK). Cecha ta zwiększa możliwości dydaktyczne AC 805, jako narzędzia wspomagającego laboratoryjną naukę techniki mikroprocesorowej.

Prezentowany komputer jest wersją podstawową z rodziny AC 800, w ramach której w roku 1985 planowane jest uruchomienie produkcji wersji **AC 825**, współpracującej z dyskiem typu **WINCHESTER-AC 925 WD.** Trwają też prace nad rozpoczęciem produkcji wersji AC 815, współpracującej ze stacjami dysków elastycznych 5,25″ — **AC 815 FDD.**

Pamięci zewnętrzne będą budowane w oparciu o napędy firmy BASF, a kontrolowane przez system operacyjny kompatybilny z CP/M 2.2. W wersji 815 użytkownik będzie miał do dyspozycji łącznie 1 MB pamięci masowej (na dwóch dyskach, przy zapisie dwustronnym). Dla modelu 825 planujemy zastosowanie dysków WINCHESTER 10 i 30 MB.

Nr programu	Czas realizacji
1	1,99
2	7,44
3	18,00
4	18,50
5	19,75
6	30,35
7	43,75
Suma	139,78

Wykonaliśmy dla AC 805 testy opublikowane w trzecim tegorocznym numerze INFORMATYKI (za PPC Computer Journal V1N1010, 1982). W tabeli podano czasy wykonywania poszczególnych programów: Suma punktów wskazuje, że AC 805 dorównuje szybkością działania podobnym komputerom tej klasy produkowanym na świecie.

RYSZARD STEFANOWSKI
AMEPROD

[1]) Konsekwencją tego może być zauważalne migotanie obrazu — w przypadkach gdy wymieniana jest znaczna część informacji zawartej w pamięci obrazu. Poza niektórymi bardzo specyficznymi programami, użytkownik nie spotyka się ze wspomnianym objawem (przypis AJP)

[2]) Zestaw laboratoryjny ZLA 01 jest jednopłytowym mikrokomputerem zbudowanym z wykorzystaniem procesora INTEL 8080 (MCY 7880) produkowanym przez P.Z. AMEPROD (przypis autora).

Tekst został opublikowany w porozumieniu z producentem, a także z finansowym wsparciem z jego jego strony. *EO/1483/K/84*

- Ponad 300 członków zrzesza węgierski klub mikro o nazwie HCC (od angielskiego: Homebrew Computer Club). W ramach klubu działa wiele sekcji zainteresowań (m.in. SINCLAIR, APPLE, VC20, RRS 80). Klub nie tylko pomaga swoim członkom w zdobywaniu wiedzy mikroinformatycznej, ale również — w zdobywaniu części zamiennych i układów.

Zamieszczając pierwszy tekst o jakimś mikrokomputerze patrzymy na niego z perspektywy potencjalnego nabywcy. Dopiero w następnych publikacjach można znaleźć bardziej szczegółowe informacje, adresowane już do użytkowników. TI 99/4A jeszcze nie opisywaliśmy. Ponieważ jednak nie będzie już nam w kraju przybywać użytkowników tego mikrokomputera (TEXAS INSTRUMENTS zaprzestał jego produkcji), publikujemy materiał użyteczny dla tych, którzy już go mają. Chociaż nie tylko dla nich. Wiele osób przymierza się do własnych 16-bitowych konstrukcji, ciułając z trudem zebrane „zielone". Być może przedstawione niżej kompromisowe rozwiązanie stanie się źródłem inspiracji — sprowadzającej konstrukcję do poziomu zasobności kieszeni. Nawet w 16-bitowych systemach pamięć jest bowiem najdroższym elementem.

Organizacja pamięci wewnętrznej mikrokomputera TI 99|4A

TI 99/4A jest zmodernizowaną wersją znanego już wcześniej mikrokomputera TI 99/4. Zasadnicza idea konstrukcji pozostała niezmieniona. Modyfikacje dotyczyły wzbogacenia mikrokomputera oraz ułatwienia jego obsługi. Również przy konstruowaniu uproszczonej wersji (TI 99/2) wykorzystano wiele rozwiązań stosowanych w poprzednich systemach.

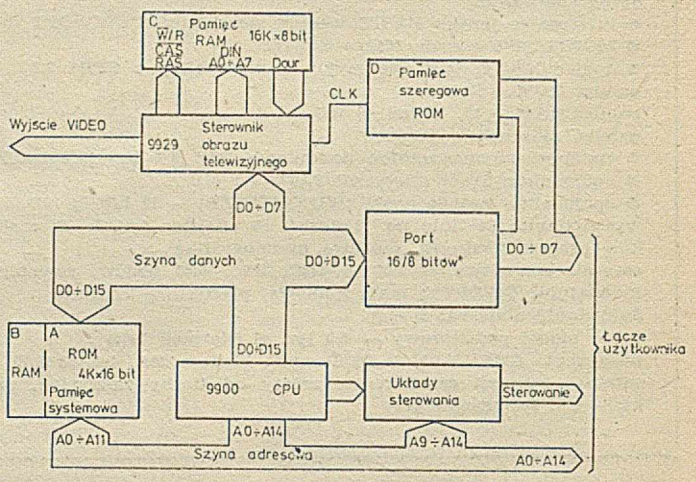
Wśród zalet systemu TI 99/4A szczególnie istotne wydają się następujące cechy:
— stosunkowo niska cena (ok. 150 dol.[1]))
— wbudowany interpreter języka BASIC (wersja TI BASIC z możliwością rozszerzenia) wzbogaconego o instrukcje graficzne, kontroli manipulatorów i sterowania dźwiękiem
— dobra klawiatura o standardowych rozmiarach i konstrukcji.
Analizując podzespoły wykorzystane w konstrukcji mikrokomputera, warto podkreślić fakt użycia procesora 16-bitowego, a więc mającego większe możliwości przetwarzania w porównaniu do procesorów o mniejszej długości słowa. Nietypowa struktura wewnętrzna procesora (organizacja typu pamięć-pamięć, bez rejestrów wewnętrznych) umożliwia łatwą zmianę kontekstu programowego i obsługę przerwań. Mimo że sam procesor został opracowany w 1976 roku, to jego parametry nie odbiegają od parametrów procesorów obecnie produkowanych (czas wykonania instrukcji dodawania — 4,7 μs dla częstotliwości zegara 3MHz). Do jego zalet należy zaliczyć osobne wyprowadzenie szyny danych i szyny adresowej oraz stosunkowo dużą liczbę linii sterujących.

Przy potencjalnie dużych możliwościach systemu, zaskakuje jednak powolność realizacji programów obliczeniowych.

Ewentualna rozbudowa systemu jest zasadniczo możliwa tylko w oparciu o moduły i urządzenia dostarczane przez producenta. Wiąże się to z niestandardowym rozwiązaniem złączy systemu oraz zastrzeżeniem producenta, iż współpraca z zewnętrznymi urządzeniami jest możliwa tylko po dołączeniu modułów programowych (w postaci pamięci ROM), współpracujących z programem nadzorującym pracę systemu. Oczywiście, użycie modułów produkowanych przez firmę TEXAS INSTRUMENTS jest rozwiązaniem najprostszym i optymalnym z punktu widzenia poprawności współpracy wszystkich urządzeń. Jednak mają one poważną wadę — koszt wielu modułów jest większy od kosztu mikrokomputera w wersji podstawowej (np. przystawka sterująca RS232C — ok. 200 dol., sterownik dysków elastycznych — ok. 220 dol.). Zapewne większość hobbystów będzie starała się rozbudowywać system w oparciu o własne pomysły i konstrukcje.

[1]) W związku z zaprzestaniem produkcji mikrokomputera przez TI, w krótkim okresie można go było nabyć — w ramach wyprzedaży — za 50 dol. — przyp. AJP.

Zasadniczą sprawą przed przystąpieniem do prac projektowych, wydaje się poznanie sposobu organizacji pamięci wewnętrznej w systemie. Struktura ta jest przykładem zastosowania procesora 16-bitowego w środowisku 8-bitowym.



Struktura pamięci wewnętrznej mikrokomputera TI 99/4A

Schemat logiczny organizacji pamięci wewnętrznej przedstawiony został na rysunku 1. Możemy wyróżnić następujące elementy pamięci:
A — blok pamięci ROM o pojemności 2 × 4 KB
B — blok pamięci RAM o pojemności 2 × 128 bajtów
C — główny blok pamięci RAM o pojemności 8 × 16 Kb
D — blok pamięci ROM o dostępie szeregowym.
Poszczególne bloki pamięci pełnią różne funkcje w systemie. Główny blok pamięci ROM (A) zawiera program nadzorujący pracę systemu (MONITOR) oraz część interpretera języka BASIC. Zorganizowany jest on w słowa 16-bitowe i współpracuje bezpośrednio z szynami: danych i adresową (przestrzeń adresowa 4 K słów 16-bitowych); współpracuje on także z pomocniczą pamięcią RAM (blok B), również zorganizowaną w słowa 16-bitowe — dwa układy 6810 produkcji firmy MOTORQLA. Wymienione bloki jako jedyne są bezpośrednio dołączone do szyn: adresowej i danych.

Pozostała część pamięci ma organizację bajtową. Najbardziej oryginalnym elementem jej organizacji jest sposób dołączenia głównego bloku pamięci RAM (C). Składa się on z ośmiu układów pamięci dynamicznych typu 4116 (w sumie 16 KB RAM). Blok C pozostaje pod całkowitą kontrolą układu TMS 9929 (będącego zarazem generatorem obrazu telewizyjnego). Układ ten dostarcza do pamięci RAM dane, sygnały adresowe, a także steruje odświeżaniem pamięci (zgodnie z własnym zegarem, stabilizowanym kwarcem). Procesor komunikuje się z pamięcią wyłącznie za pośrednictwem układu TMS 9929, korzystając z ośmiu bardziej znaczących bitów szyny danych. Systemowa szyna adresowa nie jest w ogóle doprowadzona do tego bloku pamięci. Część RAM-u jest wykorzystywana jako pamięć obrazowa, co zawęża obszar pamięci dostępny dla programów. Procesor ma dostęp do pamięci tylko w momentach, gdy nie jest ona odświeżana lub nie jest wykorzystywana do tworzenia obrazu na monitorze telewizyjnym (przy zapełnieniu ekranu zostawia to zaledwie ok. 10% czasu na dostęp do pamięci przez mikroprocesor). Sposób dostępu procesora do pamięci RAM tłumaczy znaczne spowolnienie wykonywania programów obliczeniowych w stosunku do czasu obliczonego na podstawie cykli instrukcji i okresu zegara.

Pozostała część pamięci (ROM szeregowa oraz złącze użytkownika, umożliwiające rozbudowę pamięci) dołączona jest do systemu przez bufor dwukierunkowy umożliwiający przesyłanie słowa 16-bitowego w postaci dwóch kolejnych bajtów. Rozbudowa pamięci RAM powinna więc być dostosowana do systemu 8-bitowego. Bufor zbudowany jest w oparciu o układy TTL i sterowany również układami TTL. Tak więc czasy przełączania są znacznie krótsze od okresu

zegara systemowego. Do budowy wykorzystano układy 74LS245 bardziej znacząca część szyny danych) oraz 74LS373 i 74LS244 mniej znaczący bajt szyny danych).

Pamięć ROM o dostępie szeregowym (D) współpracuje z układem TMS 9929 (pobierany jest z niego zegar przesuwający) i zawiera przede wszystkim generatory znaków graficznych (standardowych lub specjalnych) oraz procedury opisywane w specjalnym języku GROM (Graphic Read Only Memory Language). Większość modułów programowych dołączanych z zewnątrz również zawiera dodatkowe układy pamięci ROM szeregowej.

Z opisu organizacji pamięci wewnętrznej mikrokomputera TI99/4A wynika, że bez rozbudowy systemu nie jest możliwe zwiększenie szybkości przetwarzania. Ponadto komunikacja z urządzeniami zewnętrznymi w trybie przesyłania równoległego wymaga również dostosowania się do specyfiki systemu. Wydaje się, że istnieją co najmniej trzy sposoby rozbudowy bez angażowania zbyt dużych środków, szczególnie dewizowych:

- rozszerzenie pamięci RAM (sterowanej w klasyczny sposób przez szynę adresową), co jednak pociąga za sobą konieczność dołączenia pamięci szeregowej ROM, zmieniającej opis konfiguracji systemu
- w oparciu o wyprowadzone linie kontrole, danych i adresowe — dołączenie prostego układu z pamięcią EPROM przejmującego kontrolę nad programem nadzorującym
- dołączenie układu pośredniczącego do istniejącego podsystemu przesyłania szeregowego w celu sterowania i współpracy ze standardowymi urządzeniami we-wy dostosowanymi do transmisji szeregowej.

Każdy z wymienionych sposobów zmusza albo do dostarczenia programowi nadzorującemu danych w wymaganej przez niego postaci (konieczność użycia pamięci ROM szeregowej), albo też do przejęcia kontroli nad tym programem (w trybie przerwania lub przez programową zmianę kontekstu). Zamierzeniem niżej podpisanego jest opracowanie urządzenia pośredniczącego, umożliwiającego dołączanie modułów produkowanych przez krajowych wytwórców, jednakże stan zaawansowania pracy nie umożliwia jeszcze pełnej prezentacji.

LESZEK KAMIONKA
Warszawa

Sterowanie napędami dysków elastycznych (2)

Układ WD 1797, proponowany do zastosowania w konstrukcji sterownika napędów dyskowych (mikroKLAN 10), ma pięć rejestrów wewnętrznych wybieranych liniami adresowymi A_1 i A_0 (tab. 1).

Tabela 1

A_1	A_0	ODCZYT RD/=0	ZAPIS WR/=0
0	0	REJESTR STATUSU	REJESTR ROZKAZÓW
0	1	REJESTR ŚCIEŻKI	REJESTR ŚCIEŻKI
1	0	REJESTR SEKTORA	REJESTR DANYCH
1	1	REJESTR DANYCH	REJESTR DANYCH

Wpisując odpowiednie polecenie do rejestru rozkazów — sterujemy wykonywanymi przez układ zadaniami.

RESTORE (pozycjonowanie)

Po otrzymaniu tego rozkazu, FDC 1797 ustawia głowicę zapisująco-odczytującą nad ścieżką 00. Format rozkazu jest następujący (od MSB):

0 0 0 0 h v r_1 r_0

h = 1 gdy przed wykonaniem rozkazu ma nastąpić docisk głowicy R/W do nośnika magnetycznego. Jeśli v = 1, to wykonywany jest odczyt adresu z dysku i dodatkowo sprawdzenie numeru ścieżki w polu adresowym sektora. Bity r_0, r_1 — określają czas pomiędzy poszczególnymi impulsami krokowymi (tab. 2).

Tabela 2

r_1	r_0	Czas przesuwania głowicy	Czas pomiędzy kolejnymi impulsami krokowymi
0	0	3 ms (MAXI)	6 ms (MINI)
0	1	6 ms (MAXI)	12 ms (MINI)
1	0	10 ms (MAXI)	20 ms (MINI)
1	1	15 ms (MAXI)	30 ms (MINI)

MAXI Fclk = 2 MHz
MINI Fclk = 1 MHz

Po zakończeniu wykonywania tego rozkazu układ generuje przerwanie. Odczytując rejestr statusu, możemy sprawdzić poprawność wykonania operacji.

SEEK TRACK (szukaj ścieżki)

Rozkaz ten ustawia głowicę R/W na ścieżce, której numer powinien zostać zapisany w rejestrze danych jako parametr. Układ WD1717 może współpracować z napędami dyskowymi, które mają do 255 ścieżek. Format rozkazu jest analogiczny jak dla RESTORE, a tylko bit czwarty ma wartość "1".

STEP (wykonaj krok)

Po otrzymaniu tego rozkazu układ wysyła jeden impuls (powodujący wykonanie kroku) po linii „step". Wykonanie rozkazu nie zmienia stanu logicznego na linii "direction". Po opóźnieniu określonym przez bity r_1, r_0, jeśli v = 1, wykonywany jest odczyt numeru ścieżki z dyskietki. Jeśli u = 1, to aktualizowana jest wartość Rejestru Ścieżki. Po zakończeniu wykonywania rozkazu generowane jest przerwanie. Format rozkazu jest następujący (od MSB):

0 0 1 u h v r_1 r_0

STEP IN (wykonaj krok do środka)

Po otrzymaniu tego rozkazu układ wysyła impuls kroku, który przesuwa głowicę na następną ścieżkę w kierunku osi obrotu dyskietki. Jeśli u = 1, to zawartość Rejestru Ścieżki jest zwiększana o 1. Pozostałe znaczniki (h, v, r_1, r_0) mają takie same znaczenie jak poprzednio. Format rozkazu:

0 1 0 u h v r_1 r_0

STEP OUT (wykonaj krok na zewnątrz)

Układ wysyła impuls kroku, który przesuwa głowicę na poprzednią ścieżkę (czyli od osi obrotu). Jeśli u = 1, zawartość Rejestru Ścieżki jest zmniejszana o jeden. Format rozkazu:

0 1 1 u h v r_1 r_0

READ SECTOR (czytaj sektor)

Po otrzymaniu tego rozkazu układ wprowadza pole identyfikujące każdego sektora, aż do momentu pobrania prawidłowego numeru ścieżki, sektora, i strony. Wprowadzane jest również CRC. Jeżeli nie zostanie wykryta nieprawidłowość, układ po wprowadzeniu pierwszego bajtu z pola danych ustawia bit w słowie statusu oraz poziom wysoki na linii DRQ, po odczytaniu każdego bajtu z pola danych. Linia DRQ jest ustawiana w stan logiczny 1. Sygnał DRQ ponownie przechodzi w stan 0 w momencie odczytu danej

z Rejestru Danych. Jeśli w odpowiednim czasie nie nastąpi odczyt danej, to w słowie statusu jest ustawiony bit "LATE DMA". Wykonanie rozkazu kończone jest przerwaniem. Format rozkazu:

1 0 0 m F_2 E F_1 0

m = 0 odczyt jednego sektora; m = 1 — odczyt kilku sektorów; F_2 — wskaźnik długości sektora; E = 1 — 15 ms opóźnienia (CLK = 2 MHz); E = 0 — bez opóźnienia; F_1 — wskaźnik wyboru strony: F_1 = 0 — ustawia linię SSO na "0", F_1 = 1 — ustawia linię SSO na "1".

Tabela 3

Typ	Czas pomiędzy zgłoszeniami DRQ (μs)	Czas od DRQ do	
		odczytu (μs)	zapisu (μs)
5,25" FM	64	55	47
5,25" MFM	32	27,5	23,5
8" MFM	16	13	11

Maksymalne czasy przeznaczone na obsługę zgłoszenia DRQ dla zapisu i odczytu różnych dyskietek zostały zestawione w tabeli 3. Przy odczycie kilku sektorów (m = 1), wprowadzanie jest kontynuowane aż do momentu wpisania rozkazu FORCE INTERRUPT do Rejestru Rozkazów.

WRITE SECTOR (zapisz sektor)

Rozkaz ten wykonywany jest podobnie do rozkazu odczytu sektora aż do momentu odnalezienia ID odpowiedniego sektora. Następnie wykonywany jest zapis zawartości sektora na dysk. Format rozkazu:

1 0 1 m F_2 E F_1 a_0

Wskaźniki m, F_2 E i F_1 mają znaczenie identyczne jak poprzednio. a_0 = 0 — ważny sektor danych (znacznik adresowy Pola Danych — FB_H); a_0 = 1 — usunięty sektor danych (znacznik Adresowy Pola Danych = $F8_H$).

READ ADDRESS (odczytaj adres)

Format rozkazu:

1 1 0 0 0 E 0 0

Wskaźnik E ma znaczenie jak poprzednio. Układ odczytuje 6-bajtową zawartość pola ID. W kolejności odczytane są: numer ścieżki, numer strony, numer sektora, długość sektora, CRC 1 bajt, CRC 2 bajt.

READ TRACK (odczytaj ścieżkę)

Format rozkazu:

1 1 1 0 0 E 0 0

Układ odczytuje zawartość całej ścieżki — począwszy od poprzedniego zbocza sygnału "INDEX PULSE". Odczyt kontynuowany jest aż do pojawienia się po raz drugi sygnału "INDEX PULSE".

WRITE TRACK (zapisz ścieżkę)

Format rozkazu:

1 1 1 1 0 E 0 0

Rokaz ten jest wykorzystywany w momencie formatowania dyskietki. CPU musi wpisać do układu (w odpowiedniej formie) zawartość całej ścieżki, bajt po bajcie, w odpowiedzi na żądanie zgłaszane linią DRQ.

FORCE INTERRUPT (wymuś przerwanie)

Format rozkazu:

1 1 0 1 I_3 I_2 I_1 I_0

Bity I_0...I_3 definiują warunki kiedy ma być generowane przerwanie:
I_0 = 1 — w momencie przejścia linii READY z poziomu niskiego na wysoki
I_1 = 1 — z poziomu wysokiego
I_2 = 1 — w momencie pojawienia się impulsu "INDEX"
I_3 = 1 — natychmiast po odebraniu tego rozkazu.
Jeśli I_3...I_0 = 0 to układ jest zerowany bez zgłaszania przerwania.

REJESTR STATUSU

Zawartość tego rejestru pozwala stwierdzić poprawność wykonania rozkazu. Opis poszczególnych bitów Rejestru Statusu został przedstawiony w tabeli 4.

Tabela 4

Bit	Znaczenie
7	Gotowość napędu dyskowego (zanegowany stan linii „READY")
6	Zapis wzbroniony (zanegowany stan linii WRTPRT/)
5	Stan 1 — gdy głowica dociśnięta do nośnika (iloczyn logiczny HLD i HLT)
4	Stan 1 oznacza, że ścieżka, sektor lub numer strony — nie zostały odnalezione
3	Błąd CRC w polu ID
2	TRK00 — przy rozkazach przesunięcia głowicy „LOST DATA" — brak odpowiedzi na sygnał DRQ w odpowiednim czasie dla pozostałych rozkazów
1	DRQ — zgłoszenie żądania odczytu—zapisu INDEX — dla rozkazów przesunięcia głowicy
0	Zajętość — oznacza, że układ jest w trakcie wykonywania rozkazu

OPROGRAMOWANIE

Program sterujący został umieszczony w pamięci od adresu 8000H. Adresy i nazwy symboliczne poszczególnych rejestrów zostały zdefiniowane następująco:

```
COMREG   EQU   0E000H
TRKREG   EQU   0E001H
SECREG   EQU   0E002H
DATREG   EQU   0E003H
SELREG   EQU   0E004H
STANREG  EQU   SELREG
STATREG  EQU   COMREG
```

Aby odczytać lub zapisać sektor, należy ustawić parametry dla tych operacji. Parametry przenoszone są przez rejestry wewnętrzne mikroprocesora. I tak:
Rejestr B — 01 — odczyt sektora; 02 — zapis sektora
Rejestr C — bity b_1 i b_0 określają numer kieszeni napędu: sekwencja 00 — kieszeń 0, sekwencja 01 — kieszeń 1 itd. Gdy b_4 = 1, realizowane jest kodowanie FM, a gdy b_4 = 0 — kodowanie MFM. Bit b_5 = 1 określa współpracę z dyskietkami 5,25", natomiast b_5 = 0 z dyskietkami 8".
W rejestrze D podawany jest numer ścieżki. **W rejestrze E** — numer sektora. W parze **rejestrów HL** podawany jest adres DMA, czyli adres, z którego mają być pobrane dane do zapisu lub gdzie wpisano odczytane dane. Natomiast po zakończeniu realizacji rozkazu, gdy A = 00, to operacja została wykonana poprawnie, natomiast gdy A = FF_H, to operacja została wykonana błędnie.
Podprogram zapisu sektora na dyskietkę może wyglądać następująco (w parze rejestrów HL znajduje się adres obszaru pamięci zawierającego dane przeznaczone do zapisu):

```
WRTFLP: MOV   A,E        ; numer sektora
        STA   SECREG
        LXI   B,DATREG   ; adres portu danych
        LXI   D,STANREG  ; adres zewn. statusu
        MVI   A,0ACH
        STA   ODMREG     ; wpisz rozkaz do FDC
MLLP:   LDAX  D
        RRC              ; czy DRQ = 1?
        JNC   MLLP
        MOV   A,M        ; kolejny bajt danych
        STAX  B
        INX   H          ; zwiększ adres DMA
        JMP   MLLP
```

Zapis realizowany jest w pętli, aż do momentu wygenerowania przerwania kończącego wykonywanie podprogramu zapisu sektora. Podprogram obsługi przerwania wygląda następująco:

```
INTRUT: POP   PSW        ; powrót nastąpi w miejsce skąd wywołano
                         WRTFLP
        LDA   STATREG
```

```
        DB      0EDH,56H  ; kody informujące o za-
                          ; kończeniu obsługi przer-
                          ; wania
        EI
        RET
```

Podobnie można napisać pozostałe procedury obsługi operacji dyskowych. Na wydruku 1 przedstawiono kod oprogramowania umieszczonego w pamięci od adresu 8000H.

```
8000 3E D0 32 00 E0 3A 38 00
8008 32 3C 80 3E C3 32 38 00
8010 E5 2A 39 00 22 3D 80 21
8018 66 81 22 39 00 F3 ED 56
8020 E1 3A 00 E0 FB CD 40 80
8028 F3 ED 46 F5 3A 3C 80 32
8030 38 00 E5 2A 3D 80 22 39
8038 00 E1 F1 C9 00 00 00 0C
8040 78 B7 C8 AF 32 DC 80 E5
8048 D5 16 00 79 E6 03 5F 21
8050 AB 80 19 D1 79 E6 F0 B6
8058 4F 32 6E 81 32 04 E0 E1
8060 C5 01 00 1C 0B 78 B1 C2
8068 64 80 C1 3A 6E 81 4F CD
8070 00 81 E6 80 C2 6B 80 3A
8078 01 E0 FE FF CA 91 80 BA
8080 CA B0 80 E5 D5 C5 CD 0A
8088 81 C1 D1 E1 E6 98 CA B0
8090 80 E5 D5 C5 CD 50 81 C1
8098 D1 E1 3A DC 80 3C 32 DC
80A0 80 FE 0A DA 83 80 3E FF
80A8 B7 37 C9 01 02 04 08 00
80B0 78 FE 02 CA DE 80 3A 6E
80B8 81 4F E5 D5 C5 CD 36 81
80C0 C1 D1 E1 E6 9C C8 E6 10
80C8 C2 91 80 3A DC 80 3C 32
80D0 DC 80 FE 0A DA 83 80 3E
80D8 FF B7 37 C9 00 00 3A 6E
80E0 81 4F E5 D5 C5 CD 1C 81
80E8 C1 D1 E1 E6 FC C8 E6 10
80F0 C2 91 80 3A DC 80 FE 04
80F8 DA 91 80 3E FF B7 37 C9
8100 E5 D5 C5 CD 0A 81 C1 D1
8108 E1 C9 CD 5A 81 7B 32 02
8110 E0 7A 32 03 E0 3E 1F 32
8118 00 E0 FB 76 7B 32 02 E0
8120 01 03 E0 11 04 E0 3E AC
8128 32 00 E0 1A 0F D2 2B 81
8130 7E 02 23 C3 2B 81 7B 32
8138 02 E0 11 03 E0 01 04 E0
8140 3E BC 32 00 E0 0A 0F D2
8148 45 81 1A 77 23 C3 45 81
8150 CD 5A 81 3E 0F 32 00 E0
8158 FB 76 C5 01 F4 01 0B 78
8160 B1 C2 5E 81 C1 C9 F1 3A
8168 00 E0 ED 56 FB C9 00 00
8170 00 00 02 02 00 02 02 07
```

Kod programu sterującego operacjami dyskowymi

W zależności od typu stosowanego napędu i rodzaju zapisu, należy przyjąć:
— częstotliwość zegara 2 MHZ (Z80 CPU) dla 5,25″ i pojedynczej gęstości
— częstotliwość zegara 4 MHz (Z80A CPU) dla 5,25″ i podwójnej gęstości zapisu lub 8″ i pojedynczej gęstości zapisu
— częstotliwość zegara 6 MHz (Z80B CPU) dla 8″ i podwójnej gęstości zapisu.

MIROSŁAW PACZEŚNY
CSK Gdynia

Co? Gdzie?

Podczas 56 Międzynarodowych Targów Poznańskich (p. str. 24) można było znaleźć wiele towarów ułatwiających pracę przy budowie, uruchamianiu i naprawach sprzętu cyfrowego, zwłaszcza mikrokomputerów. Oto ciekawsze z nich, opisane na podstawie materiałów firmowych:

Impulsatory logiczne

Przypominają one znane w krajach zachodnich urządzenia „current pulser" do pobudzania sieci układów logicznych, bez rozcinania połączeń na płytkach drukowanych. Impulsatory logiczne umożliwiają krótkotrwałe (200—700 ns) wymuszenie nowego stanu logicznego, niezależnie od sposobu połączenia sieci logicznej. Skutki tej chwilowej zmiany można zaobserwować za pomocą sondy logicznej, pozwalającej na wydłużenie impulsów.

Logiczny Próbnik Impulsator typu LPI-145 nie tylko umożliwia wymuszenia zmiany stanu logicznego, ale może również zastąpić sondę logiczną. Informacje: Przedsiębiorstwo Techniczno-Handlowe KABIDEZ; 03-468 Warszawa, ul. Stalingradzka 29/31; tel. 11-08-48.
Impulsator Logiczny typu LI-SI uzupełnia sondę logiczną typu SLS-1. Informacje: Zakłady Urządzeń do Montażu Podzespołów Elektronicznych UNITRA-CEMI; 12-100 Szczytno, ul. Dąbrowskiego 6; tel 32-81.

Analizator stanów logicznych

Analizator typu E-220 charakteryzuje się częstotliwością próbkowania do 20 MHz i pozwala na wykrywanie krótkich impulsów (ok. 10 ns). Przyjmuje sygnały o poziomach TTL i ma osiem wejść danych. Zapis jest asynchroniczny lub synchroniczny; można dokonać wyboru zbocza aktywnego. Sygnał zegarowy — zewnętrzny lub wewnętrzny (20 MHz-50 Hz). Zapisywanie danych pod kontrolą sygnałów START i TRIGGER, który można określać jako 8--bitowe słowo. Dane mogą być rejestrowane przed i po wyzwoleniu, a opóźnienie zapisu można nastawiać w zakresie 0—9999 okresów zegara. Pamięć — 256 bajtów. Wyświetlanie na oscyloskopie lub monitorze telewizyjnym. Informacje: Przedsiębiorstwo Doświadczalno-Produkcyjne Elektronicznej Aparatury Pomiarowej EUREKA; 00-277 Warszawa, ul. Freta 39; tel. 31-32-85.

Zestaw odlutowniczy

Jest to narzędzie o nazwie BTT-101 przydatne do wylutowywania podzespołów elektronicznych z płytek drukowanych. Grot został tak zbudowany, że możliwe jest odsysanie lutowia lub wydmuchiwanie go z naprawianego połączenia. Dzięki wbudowanej pompie, nie jest wymagana instalacja sprężonego powietrza. Informacje: Instytut Komputerowych Systemów Automatyki i Pomiarów; 51-608 Wrocław, Al. Młodej Gwardii 1c; tel. 48-10-81 w. 256.

Koszulki izolacyjne termokurczliwe

Produkcji krajowej. Informacje: Zakład Urządzeń Technologicznych; 77-300 Człuchów, ul. Sienkiewicza 21; tel. 771.

Kasety w standardzie Eurokarty

Są one zgodne z normą SEW-3266-81 oraz DIN 41404. Mogą pomieścić 21 modułów o szerokości 20,32 mm i nadają się do montażu w stojakach 19". Istnieją trzy wersje, dla różnych wymiarów płytek drukowanych:
— 0.11.01.34 dla płytek drukowanych o wymiarach 100 × 160 mm
— 0.21.01.34 — o wym. 160 × 233,35 mm
— 0.22.01.34 — o wym. 220 × 233,35 mm
Informacje: Zakłady Urządzeń Przemysłowych POLON; 30-133 Kraków, ul. Dzierżyńskiego 124; tel. 37-39-00.

Zasilacze do systemów cyfrowych

Zasilacze stabilizowane do mikrokomputerów:
EZP 01-00 z wyjściem 5 V/40A
EZP 01-02 — 12V/20A
EZP 01-04 — 24V/10A
EZP 02-10 — 5V/20A
EZP 02-11 — 12V/10A
EZP 02-14 — 24V/5A
EZP 06-00 — 5V/20A
EZP 07-00 — 5V/40A
EZP 08-01 — 5V/10A, +12V/4A, —12V/1A
EZP 08-02 — 5V/10A, +15V/2A, —15V/2A
EZP 04-01 — 5V/20A, +12V/2A, —12V/2A
EZP 04-02 — 5V/20A, +15V/1A, —15V/1A
EZP 05-01 — 5V/20A, +12V/2A, —12V/1A, —5V 1A.

Informacje: Zakłady Automatyki Przemysłowej; 63-400 Ostrów Wielkopolski, ul. Krotoszyńska 35; tel. 624-21.

Zasilacze sieciowe stabilizowane:
SPS 1A-5.20 SC z wyjściem 5V/20A
SPS 1C-5.40 SC — 5V/40A
MPS-150-3/2 — 5V/20A, +12V/2A, —12V/2A
MPS-150-3/3 — 5V/20A, +15V/1A, —15V/1A
MPS-150-3/4 — 5V/20A, +12V/2A, —5V/1A
MPS-120-3/2 — 5V/10A, +12V/4A, —12V/1A
MPS-120-3/3 — 5V/10A, +15V/2A, —15V/2A
MPS-150-4/2 — 5V/20A, +12V/2A, —12V/1A, —5V 1A
MPS-150-4/4 — 5V/20A, +15V/1A, —15V/1A, —5V 1A.

Informacje: Zakład Doświadczalny Elektroniki i Mechaniki Precyzyjnej Politechniki Śląskiej; 44-100 Gliwice, ul. Bałtycka 8; tel. 31-80-81.

Przetworniki DC/DC

Przetworniki napięcia stałego na stałe (z izolacją galwaniczną) są przeznaczone do wytwarzania napięć pomocniczych, przydatnych, na przykład, do zasilania modułów we-wy. Są one zasilane z napięcia głównego +5 V i można je montować bezpośrednio na obwodach drukowanych. Zakład Dowiadczalny Elektroniki i Mechaniki Precyzyjnej opracował takie przetworniki, które z napięcia +5 V dają stabilizowane napięcia +12 V i —12 V lub +15 V i —15 V, o obciążalności od 100 do 200 mA na każdy biegun strony wtórnej.

Niestabilizowaną przetwornicę +5 V/±12 V o obciążalności 2 × 40 mA opracował Ośrodek Badawczo-Rozwojowy Elektronicznych Układów Specjalizowanych MERA-OBREUS; Informacje: Toruń, ul. Grudziądzka 46; tel. 330-45.

Nowoczesne środki chemiczne dla elektroniki

Preparaty klejowe do łączenia elementów wykonanych z różnych materiałów, regeneracji i korekty obwodów drukowanych, ekranowania i odprowadzania ładunków elektrostatycznych. Są to: THERMOCON — kleje elektroizolacyjne przewodzące ciepło, ELEPOX — kleje przewodzące prąd elektryczny oraz ELECTROCON — lakiery przewodzące. Informacje: Przedsiębiorstwo Polonijno-Zagraniczne AMEPOL; 01-126 Warszawa, ul. Brylowska 8a; tel. 32-73-66.

JACEK ŻEBROWSKI

Choć napływają do nas zewsząd prośby o pomoc dla mikroinformatycznych nowicjuszy, nie jesteśmy w stanie zastąpić Wydawnictw Naukowo-Technicznych. Nie zamierzamy spełniać funkcji substytutu książek, które — wierzymy — kiedyś wreszcie zostaną wydane.

Łamy mikroKLANU adresujemy do stawiających drugi i dalsze kroki. Zatem do lektury AKADEMII zapraszamy tych, którzy już przeczytali instrukcję obsługi posiadanego mikrokomputera (praktycznie zawsze można tam znaleźć lepszy lub gorszy opis zastosowanej wersji języka BASIC).

Eleganckie programowanie, które chcemy lansować w AKADEMII mikroKLANU pozwala na rzecz chyba najważniejszą — wielokrotne wykorzystanie. Jak wykazuje praktyka — program, który okazał się użyteczny, prawie zawsze będzie wymagał w przyszłości zmian. Aby można je było wprowadzić, trzeba dokładnie wiedzieć, co robią poszczególne bloki programu.

AKADEMIA mikroKLANU

Czy potrafimy programować elegancko? Elegancja wyszła jakoś z mody i w rezultacie mamy szeroką rzekę programów, które często są nierozumiałe nawet dla autorów. Ludzie wymyślili języki programowania wysokiego poziomu po to, aby programy napisane wczoraj mogły być zrozumiane jutro. Również przez innych.

Wprawdzie E. W. Dijkstra (którego książkę — „Umiejętność programowania", WNT, 1978 — wszystkim polecamy) ostro potępia BASIC jako najgorszy język programowania mikrokomputerów i nic na to nie poradzimy. Nasz kurs eleganckigeo programowania będzie więc oparty na standardowym BASICU (p. INFORMATYKA 5/83).

Dziś pierwszy odcinek, a w nim — test komputera. Niestety, nasza Akademia nie może zapewnić kursantom pomocy naukowych...

WŁAŚCIWE WYJŚCIE

Każdy kurs programowania zaczyna się od pytania: „Jaki element programu jest niezbędny?". Wszyscy kursanci zapewne krzykną tu chórem: „Wyjście!!". I będą mieli rację, choć jest to całkiem odwrotnie niż w życiu codziennym (gdy jest alternatywa, ale nie ma wyjścia). W programowaniu nie ma alternatywy — musi być wyjście, gdyż informuje nas ono o tym, **co** dany program robi (lub zrobił). Najgorszym możliwym wyjściem jest wyjście typu:

```
RUN
M= 1.23
X=—1
READY.
```

Może być ono wystarczające dla autora programu (który wie, że **M** jest liczbą nosorożców, a **X** — temperaturą), ale innemu użytkownikowi nic to nie mówi. Tym bardziej, gdy — jak w naszym przykładzie — liczba nosorożców wychodzi niecałkowita, a temperatura ujemna. Takie wyjście nie tylko nie pozwala sprawdzić, czy program działa poprawnie, ale co gorzej — przekazuje informację niepełną, czyli dezinformuje. Co należy robić, aby wyjście było czytelne? Pisać, pisać, pisać! Najlepiej pełnymi zdaniami, bez skrótów i uproszczeń. Zaprocentuje nam to już za pół roku, gdy bez kłopotów będziemy rozumieli własny program.

Pierwszy „szkolny" program przedstawiono na wydruku.

```
LIST

5 REM PROGRAM TESTUJACY DOKLADNOSC
6 REM OBLICZEN WEWNETRZNYCH KOMPUTERA.
10 N=-2
20 X=1
30 Y=81
40 Y=X/Y
50 Y=Y*10
60 X=INT(Y)
70 READ A
80 N=N+1
90 Y=Y-A
100 IF X=A THEN GOTO 50
110 PRINT "TEN KOMPUTER WYKONUJE OBLICZENIA NA ";N;" MIEJSCACH DZIESIETNYCH"
120 DATA 0,1,2,3,4,5,6,7,9,0,1,2,3,4,5,6,7,9,0,1
130 END
READY.
     RUN
NEPTUNE 184 WYKONUJE OBLICZENIA NA  10  MIEJSCACH DZIESIETNYCH

READY.
```

Zwróćmy uwagę, że każdy program powinien zawierać na początku choć jedną linijkę komentarza (u nas linie 5 i 6), objaśniającego **co** program robi — samo wyjście może być niezbyt zrozumiałe!

Parę uwag praktycznych. Dla większej przejrzystości opuszczamy w naszych programach słówko LET — w rzeczywistości wiele interpreterów BASIC wymaga podania tego słowa w tekście programu (np. ZX SPECTRUM). Każdy program powinien mieć też zakończenie. W niektórych wersjach języka BASIC (jak we wspomnianym już SPECTRUM) nie ma instrukcji END — wtedy należy użyć instrukcji STOP. Przestrzeganie tego polecenia stanowi przynajmniej częściowe zabezpieczenie przed programami zapętlonymi. Wprawdzie dla mikrokomputerów takie zapętlenie nie jest zbyt groźne (zawsze można wyłączyć komputer), ale elegancja wymaga, by użytkownik wiedział gdzie kończy się program.

Wreszcie uwaga końcowa — wszyscy dysponujący komputerami o rozszerzonym BASICU, który pozwala używać zmiennych wieloliterowych, powinni nadawać zmiennym sensowne i zrozumiałe nazwy. Coś podobnego zostało zrobione w naszym programie — N zazwyczaj oznacza liczbę zdarzeń; X, Y, Z używamy do opisu zmiennych (to pamiątka z przeszłości, gdy liczyliśmy na kalkulatorze), a stałe oznaczamy A, B, C itd.

Już słychać głośne BEEP; cóż, lekcja skończona. Do następnego spotkania!

JAKUB TATARKIEWICZ

---

prowadzi
ANDRZEJ J. PIOTROWSKI
tel. dom. 48-22-85